Collaborative Filtering using Neural Networks for Explicit Feedback Recommendation Systems

Abhineet Pandey
Department of Computer Science
Indian Institute of Technology, Ropar

Bhawna
Department of Computer Science
Indian Institute of Technology, Ropar

Dr. Shashi Shekhar Jha
Assistant Professor
Department of Computer Science
Indian Institute of Technology, Ropar

December 3, 2019

Abstract

The paper discusses building recommendation systems for explicit feedback systems where the task is to recommend items to users based on a rating matrix which is a matrix where cell (i,j) corresponds to the rating user 'i' gave to item 'j'. Based on these ratings, the system tries to predict unfilled ratings in the matrix i.e. predicts intelligently what rating value an item might get by a user. The traditional collaborative filtering technique for recommender systems depends upon the similarity values between two given user or item vectors in the rating matrix. But the technique in general suffers from some major problems such as sparseness of user profiles and scalability.[1] Deep neural networks have achieved success in various fields such as image classification, segmentation, speech recognition and and many other domains. The idea is to replace the inner product in the matrix factorization with a deep neural network.[2] The technique of neural collaborative filtering[2] has been used in systems based on implicit feedback which classifies a given item by all the users as either might be interested in or not interested in, the task in explicit feedback systems is to have techniques for getting the the exact ratings from 1 to 5 that user might give to an item. We suggest two different approaches to use Neural Networks with collaborative filtering and matrix factorization where we concatenate user and item embeddings and then develop a model over them.

1 Introduction

Recommendation systems are used to recommend items such as movies, webpages, products etc to potential customers. The advent of immense digital culture has to be supported by systems that can predict quite accurately and efficiently products to right customers/users and eventually facilitate large revenues for e-firms. There are various different techniques employed by different large firms for their products but the technique of collaborative filtering which uses aggregated behaviour of customers and items remains the most used due to its simplicity as well as lower online computation time than most other techniques which is linear in number of most similar users or items chosen. The technique has been employed in various internet companies like Amazon[3], Netflix[4], Google News[5], and others. But the technique of collaborative filtering combined with neural networks in the field of explicit feedback systems to the best of our knowledge remains largely unexplored. Some recent work in deep learning for recommendation systems mainly employs the deep learning models for textual descriptions/ features of items. The interaction between user and item features relies upon an inner product on the latent features of users and items[2].
For the purpose of recommending the items to users, we have used a model which first has an embedding layer for both users and items(movies in our case). It then concatenates the two embeddings (which function as feature vectors) for the purpose of being fed to the standard neural network with fully connected layers. We compare our results (using the

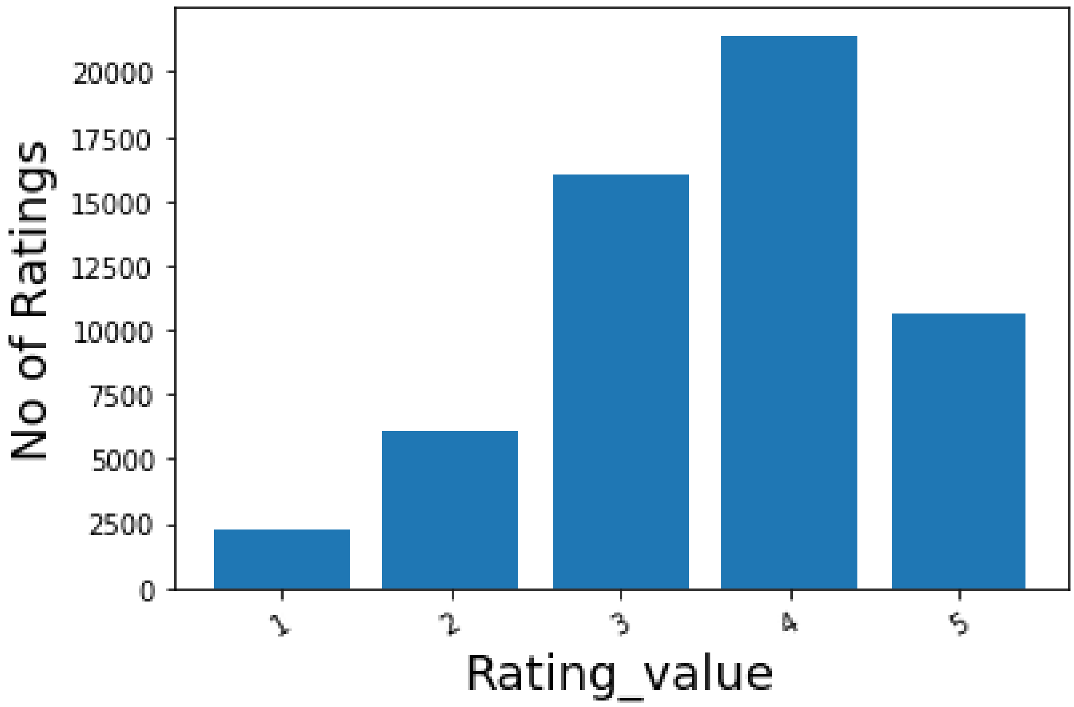

Figure 1: Distribution of ratings in the dataset.

metric of mean absolute error) on a randomly selected part of MovieLens Dataset with approximately 9000 movies and 600 users.

2 Analysis of Dataset Used

The classic MovieLens dataset has been used for predicting ratings. Number of movies and users in the dataset are 9745 and 600 respectively. There is a random disparity between number of ratings with values 5, 4, 3, 2 and 1 as shown in figure 1. The test set comprises of 20168 ratings to be predicted for 600 users for a given set of movies. The problem here like any other real world recommendation dataset is the need to predict ratings for users having rated very sparsely too as the test set needs the prediction for all users to be predicted comprehensively.

3 Architecture Employed for Neural Collaborative filtering

3.1 Approach 1

We used a tabular data method to predict the ratings. The model comprises of an embedding layer of dimension 50 each for users as well as movies. The embedding layers are then flattened. A dropout layer of drop out factor 0.2 is used for both users and movies. We first use a matrix product and flatten layer followed by a fully connected layer to get a mean absolute error of 0.94 which is far better than what achieved through traditional matrix factorization and collaborative filtering. This motivated us to use neural networks for the use of predicting the interaction between user and item (movies) latent factors. We make a modification of concatenating the user and item embeddings and then building several fully connected layers over it. For explicit feedback, we use a pointwise loss function Root Mean Squared Error instead of pairwise loss function which has been used in neural network based collaborative filtering but for the case of implicit feedback. We achieve a mean absolute error of 0.847 far better than that achieved through using neural networks only for predicting the inner product approximation through a continuous function. The architecture in whole is represented in Figure 2. We use Adam optimizer for gradient descent over the layers. The output layers give the probability outputs which are then evaluated through RMSE loss function. If y is the required rating, y_i the predicted rating and n is the number of ratings or predictions, the loss function is as defined as below:

RMSE = $\sqrt{\sum (y - \mathrm{y}_i)^2} / \sqrt{n}$

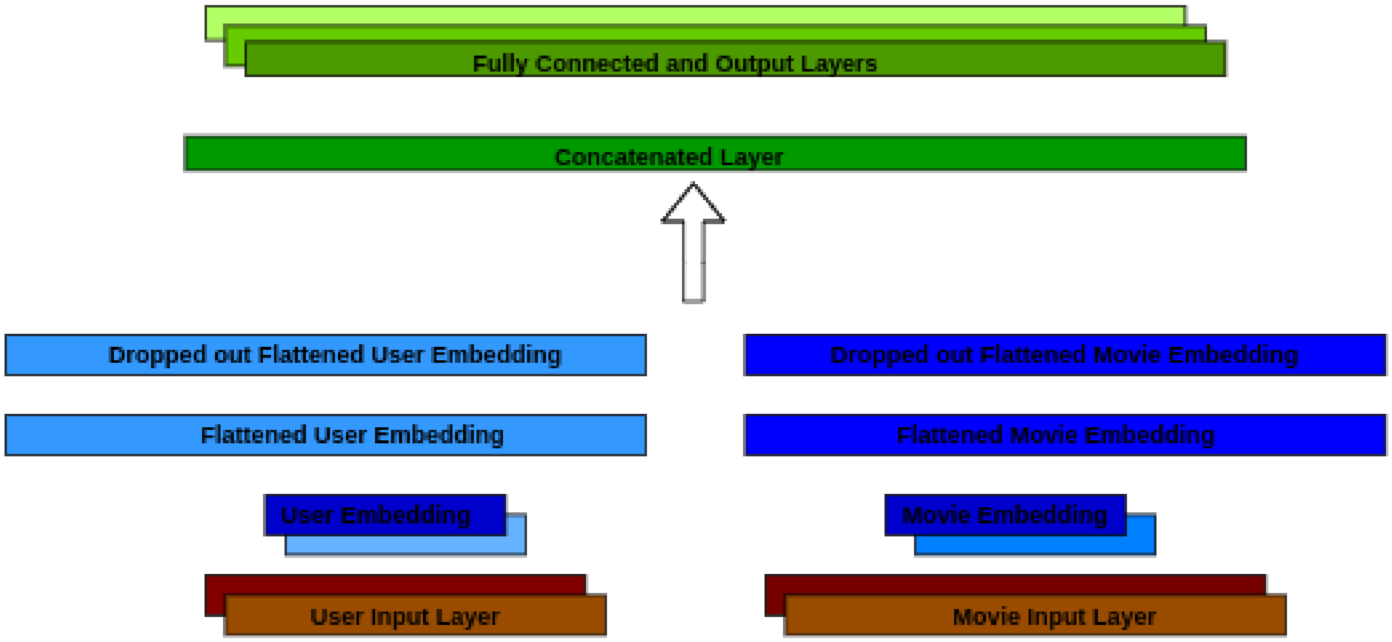

Figure 2: Approach 1 Architecture Employed for Explicit Feedback.

3.2 Approach 2: Matrix Factorization, Neural Network and Fusion

Having derived the idea from the approach suggested in Neural Collaborative Filtering[2], we develop an architecture which involves Matrix Factorization and Multi Layer Perceptron followed by a neural network. The layers obtained from both these parts of the model (matrix factorization and neural network over the user and item layer embeddings) are concatenated. This concatenated layer is followed by an output layer followed by a relu activation function as in all above models implemented. The loss function used is root mean square error instead of a pairwise loss function used in implicit feedback.
The model is as described below in Figure 3.
The mean absolute error achieved through this approach is 0.819 which is a decent improvement over the earlier approach in which we didn't combine the multi layer perceptron model which is an improvement over the earlier approach.

4 Results and Comparisons with other Techniques Implemented

For the purpose of knowing the efficiency of our approach and comparing it with the other techniques developed so far, we implemented and compared our results with the following approaches:
i) Traditional Collaborative Filtering with Mean Centred Matrix
ii) Matrix Factorization with fully connected layers for inner product calculation and final rating prediction.
iii) Our Approach 1: Used User and Item Embeddings concatenated followed by a deep neural network to get the predictions.
iv) Collaborative Filtering with user and item biases (implemented through FastAI)
v) Collaborative Filtering with neural networks (FastAI)
vi) Our Approach 2: Fusion of Matrix Factorization and Neural Network (Derived through NCF [2])
The metric used for comparing the techniques was Mean Absolute Error given by

$$MAE = \frac{\sum_{i=1}^{n}(x_i - x)}{n} \tag{1}$$

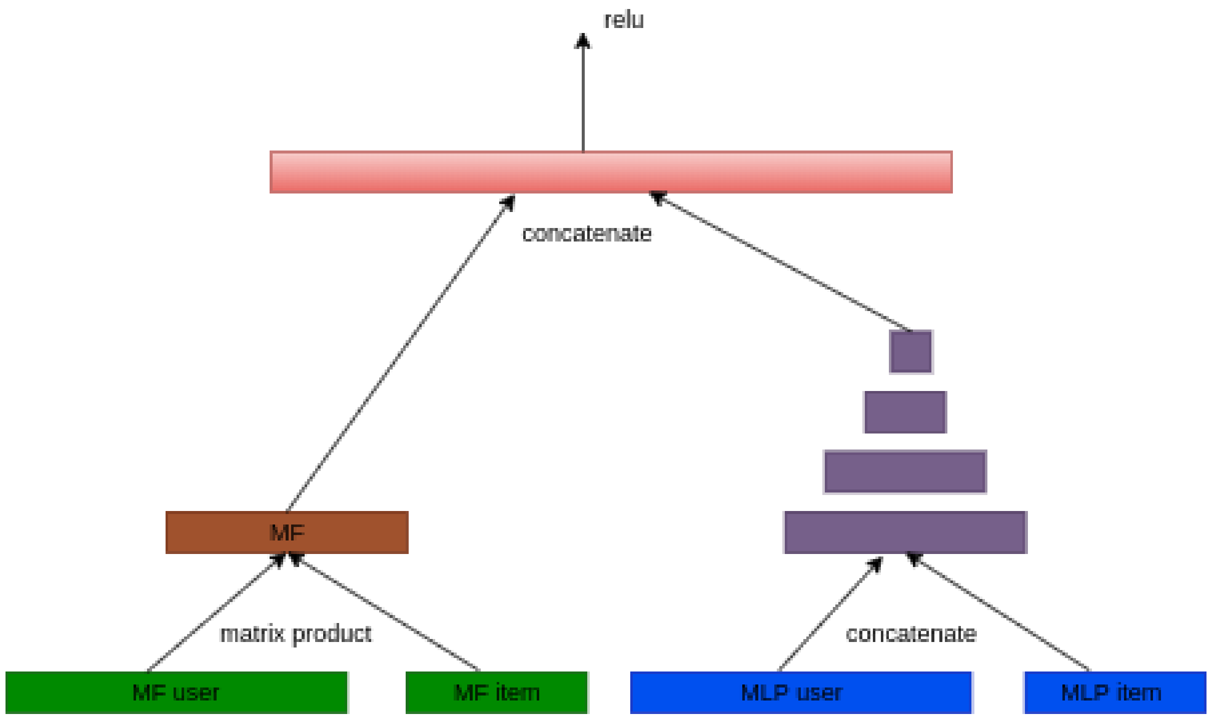

Figure 3: Approach 2 Architecture Employed for Explicit Feedback.

4.1 Collaborative Filtering

The collaborative filtering technique with Pearson Similarity Coefficient was implemented with K=100. The steps are as in the traditional collaborative filtering:
a) The rating matrix was first mean centered by subtracting row means from each entry in each row.
b) Similarity between all two users and all possible two items are calculated based on pearson similarity coefficient defined in equation below.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2}$$

c) Next top K similar users are picked for each user and a weighted average of the ratings of these users for any given item is used as a prediction for the user. The mean absolute error achieved in this case is 1.3.

4.2 Matrix Factorization with Neural Network

The implementation comprised of user and item latent factors multiplied as in Matrix Factorization.
If Rating matrix is R, user and item embeddings U and V respectively, then an intermediate matrix U*V^T is obtained. A neural network is built over this matrix product with flatten and fully connected layers. The loss function is RMSE as described above along with adam optimizer. The mean absolute error obtained with this method was 0.9383 which is still better than the mean absolute error achieved through traditional collaborative filtering which was 1.3.

4.3 Neural Collaborative Filtering (Approach 1)

The approach as explained in the architecture above comprises of layers described in Figure 2.
The mean absolute error achieved in this case is 0.847 which is a very significant improvement over Matrix Factorization in previous case which was 0.9383.

Table 1: Summary Table

Technique/Approach		
Method	Parameters	Mean Absolute Error
Collaborative Filtering	k=100	1.3
Matrix Factorization with Neural networks	k=50	0.938
Approach 1	k=50	0.847
FastAI Collaborative Filtering	k=10	0.843
FastAI Collaborative Filtering with Neural Networks	k=10	0.824
Approach 2	k=10	0.818

4.4 Collaborative Filtering with User and Item Biases

With fastAI library, collabDataBunch was used for getting input rating matrix and then collabLearner class was used with a learning rate of 2e-3. The library function gave an accuracy of 0.8430 which is approximately same as that achieved through our suggested approach.

4.5 Collaborative Filtering with Neural Networks(FastAI)

With factor dimension and embedding size 10, further layers of size 256, 128 and 64 size were used which gave a mean absolute error of 0.8241 which is slightly better than neural collaborative filtering approach suggested.

4.6 Matrix Factorization and Neural Network (Approach 2)

The approach discussed above in section 3.2 with learning rate of 1e-4 and parameter K = 10 gave an MAE of 0.818. The layer sizes in the Multi Layer Perceptron model comprises of layers with sizes 64, 32, 16 and 8.

4.7 Summary of Results

Table 1 enlists the followed approaches and corresponding results summarized.

5 Possible Future Extensions of Approach 2

The approach 2 has some issues that arise with any multi-class classification. Given that we have used ReLu in the final output layer, the output from the model is continuous. However, multi-class classification have been shown to work better in many cases using one vs all classification techniques, i.e. to use one model with sigmoid in final output layer for each possible rating, and then to choose the rating with maximum confidence. Another problem arises due to class imbalance, which can be dealt with by creating a weighted loss function, so that the model doesn't ignore the classes in the process of minimizing the objective function.

References

[1] https://link.springer.com/chapter/10.1007/978-3-540-68880-8_32

[2] https://arxiv.org/abs/1708.05031

[3] Linden, G., Smith, B., York, J.: Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing 7, 76–80 (2003)

[4] Netflix CineMatch, http://www.netflix.com

[5] Das, A., Datar, M., Garg, A., Rajaram, S.: Google news personalization: Scalable online collaborative filtering. In: Proc. of WWW 2007, pp. 271–280 (2007)